



04cd 08-15-01 #4

Docket No.: GR 98 P 5928 P

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231, on the date indicated below.

By: Markus Nolf Date: August 21, 2001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Michael Weber et al.  
Applic. No. : 09/882,730  
Filed : June 15, 2001  
Title : Method for changing the execution of a program stored in a read only memory

CLAIM FOR PRIORITY

Hon. Commissioner of Patents and Trademarks,  
Washington, D.C. 20231

Sir:

Claim is hereby made for a right of priority under Title 35, U.S. Code, Section 119, based upon the German Patent Application 19857921.7, filed December 15, 1998.

A certified copy of the above-mentioned foreign patent application is being submitted herewith.

Respectfully submitted,

Markus Nolf  
For Applicants

MARKUS NOLFF  
REG. NO. 37,006

Date: August 21, 2001

Lerner and Greenberg, P.A.  
Post Office Box 2480  
Hollywood, FL 33022-2480  
Tel: (954) 925-1100  
Fax: (954) 925-1101

/kf



# BUNDESREPUBLIK DEUTSCHLAND



## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 198 57 921.7

**Anmeldetag:** 15. Dezember 1998

**Anmelder/Inhaber:** Siemens Aktiengesellschaft, München/DE

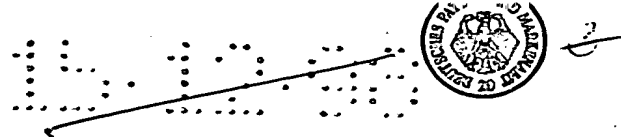
**Bezeichnung:** Verfahren zur Änderung des Ablaufs eines in einem  
Festwertspeicher abgelegten Programms

**IPC:** G 06 F 9/445

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der  
ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 7. August 2001  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

Dzierzon



### Zusammenfassung

Die Erfindung betrifft ein Verfahren zur Änderung des Ablaufs eines in einem Festwertspeicher abgelegten Programms. Das Programm weist dazu eine Vielzahl von Programmroutinen auf und jeder Programmroutine ist ein Unterprogramm, das in einem ersten Schreib/Lese-Speicher abgelegt ist, zuordenbar. Jeder Programmroutine sind weiterhin Speicherstellen eines zweiten Schreib/Lese-Speichers zugeordnet. Eine Programmroutine ruft dann, falls dieser Programmroutine ein Unterprogramm zugeordnet ist, das Unterprogramm abhängig vom Inhalt der zugeordneten Speicherstellen auf.

Figur 1

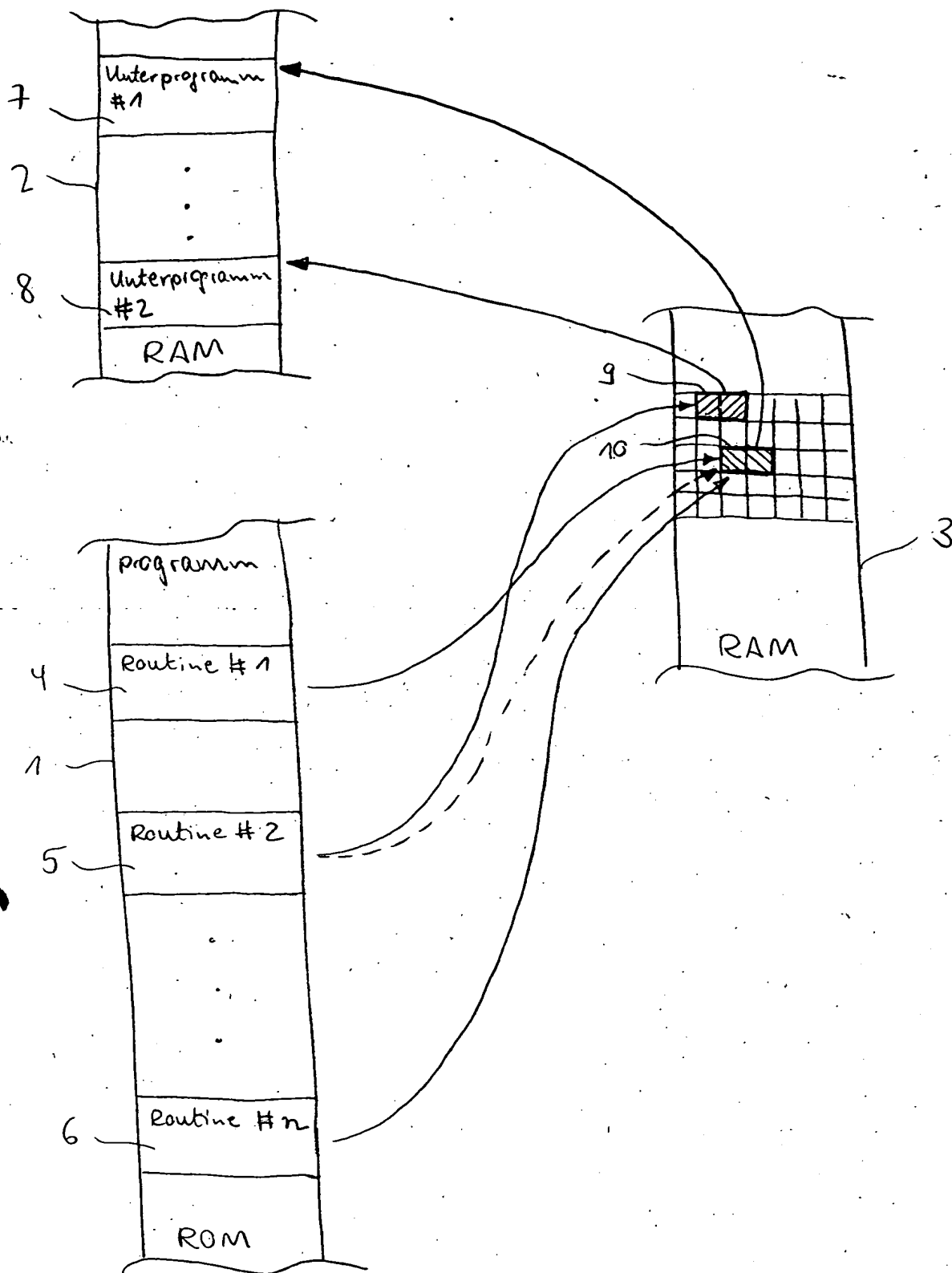


Fig. 1

## Beschreibung

Verfahren zur Änderung des Ablaufs eines in einem Festwert-  
speicher abgelegten Programms

5

Die Erfindung betrifft ein Verfahren zur Änderung des Ablaufs  
eines in einem Festwertspeicher abgelegten Programms nach dem  
Oberbegriff von Patentanspruch 1.

- 10 Viele integrierte Schaltungen, die einen Mikroprozessor  
und/oder einen Signalprozessor beinhalten, weisen einen Fest-  
wertspeicher (ROM = Read Only Memory) für Programme des Mi-  
kroprozessors bzw. Signalprozessors (Firmware) auf. Bei-  
spielsweise werden bei einer integrierten Schaltung für ein  
15 Mobilfunk-Telefon nach dem GSM-Standard in Programmen für ei-  
nen Signalprozessor Funktionen zur Sprachverarbeitung, Kanal-  
kodierung und für Datendienste und in Programmen für einen  
Mikroprozessor zur Steuerung des Telefons realisiert. Für  
nachträgliche Änderungen der Programme muß mindestens eine  
20 Maske zur Produktion der integrierten Schaltung geändert wer-  
den und die integrierte Schaltung neu produziert werden. Ins-  
besondere bei der Massenproduktion von integrierten Schaltun-  
gen ist dies aufwendig und teuer.

- 25 Aus US 5,493,674 ist beispielsweise bekannt, in einem Pro-  
gramm, das in einem Festwertspeicher abgelegt ist, an vorge-  
gebenen Stellen Befehle einzufügen, die jeweils ein Unterpro-  
gramm (Patch-Correction-Program) aufrufen. Dabei ist das Un-  
terprogramm in einem Schreib/Lese-Speicher (RAM = Random  
30 Access Memory) abgelegt und nachträglich änderbar.

- Vorteilhaft ist dabei, daß nachträglich durch Änderung der im  
Schreib/Lese-Speicher abgelegten Unterprogramme der Ablauf  
des im Festwertspeicher abgelegten Programms beeinflussbar  
35 ist. Dazu wird der Schreib/Lese-Speicher mit den Unterpro-

15.10.98

6

grammen zur Korrektur des Festwertspeicher-Programms von Vorrichtungen, die nicht auf der integrierten Schaltung enthalten sind, geladen.

- 5 Nach Ausführung des Unterprogramms wird in das im Festwertspeicher abgelegte Programm zurückgesprungen.

10 Nachteilig ist jedoch, daß vom laufenden Programm immer in die Unterprogramme gesprungen wird. Ferner ist für jeden Befehl zum Aufruf eines Unterprogramms genau ein Unterprogramm vorgesehen, wodurch sich für die Unterprogramme unter Umständen ein großer Speicherplatzbedarf ergibt. Durch diese engen Rahmenbedingungen sind nachträgliche Änderungen des im Festwertspeicher abgelegten Programms nur bedingt möglich.

15

Das der Erfindung zugrundeliegende technische Problem besteht daher darin, ein Verfahren zur Änderung des Ablaufs eines in einem Festwertspeicher abgelegten Programms anzugeben, bei dem sowohl nachträglich die Unterprogrammaufrufe beeinflussbar  
20 sind als auch der für die Unterprogramme benötigte Speicherplatzbedarf möglichst gering ist.

Dieses Problem wird von einem Verfahren zur Änderung des Ablaufs eines in einem Festwertspeicher abgelegten Programms  
25 mit den Merkmalen von Patentanspruch 1 gelöst. Vorteilhafte Ausgestaltungen der Erfindung ergeben sich aus den jeweiligen Unteransprüchen.

Die Erfindung betrifft ein Verfahren zur Änderung des Ablaufs  
30 eines in einem Festwertspeicher abgelegten Programms. Das Programm weist dazu eine Vielzahl von ProgrammROUTINEN auf und jeder ProgrammROUTINE ist ein Unterprogramm, das in einem ersten Schreib/Lese-Speicher abgelegt ist, zuordenbar. Demnach kann jeder ProgrammROUTINE ein eigenes Unterprogramm,  
35 mehreren ProgrammROUTINEN ein gemeinsames Unterprogramm oder

15. 12. 98

7

keiner Programmroutine ein Unterprogramm zugeordnet sein. Nur wenn das im Festwertspeicher abgelegte Programm fehlerfrei ist, wird dabei keiner Programmroutine ein Unterprogramm zugeordnet sein. Jeder Programmroutine sind weiterhin Speicherstellen eines zweiten Schreib/Lese-Speichers zugeordnet. Eine Programmroutine ruft dann, falls dieser Programmroutine ein Unterprogramm zugeordnet ist, das Unterprogramm abhängig vom Inhalt der zugeordneten Speicherstellen auf. Durch die einer Programmroutine zugeordneten Speicherstellen kann somit ein Unterprogrammaufruf beeinflußt werden. Der Vorteil dieses Verfahrens besteht darin, daß die Unterprogramme bedingt aufgerufen werden. Vorteilhaft ist dabei zudem, daß die Bedingung zum Aufrufen eines Unterprogramms durch die Belegung der Speicherstellen, die einer Programmroutine zugeordnet sind, jederzeit durch Umprogrammierung nachträglich geändert werden kann. Diese Verfahren zeichnet sich gegenüber bekannten Verfahren durch seine Vielseitigkeit aus.

In einer bevorzugten Ausführungsform übergibt jede Programmroutine beim Aufruf des ihr zugeordneten Unterprogramms mindestens einen Parameter an das Unterprogramm. Vorteilhafterweise wird dadurch das Verfahren noch vielseitiger, da das Unterprogramm in Abhängigkeit von dem übergebenen Parameter unterschiedliche Funktionen ausführen kann.

Die einer Programmroutine zugeordneten Speicherstellen sind in einer besonders bevorzugten Ausführungsform ausschließlich dieser Programmroutine zugeordnet. Diese Ausführungsform wird verwendet, wenn der Speicherplatzbedarf des zweiten Schreib/Lese-Speichers unerheblich ist. Von Vorteil ist dabei die große Vielseitigkeit beim Aufruf der Unterprogramme, da für jede Programmroutine durch entsprechende Belegung der zugeordneten Speicherstellen festgelegt werden kann, ob das zugeordnete Unterprogramm aufgerufen werden soll. In einer dazu alternativen besonders bevorzugten Ausführungsform sind die

15.12.98

einer Programmroutine zugeordneten Speicherstellen auch allen anderen Programmroutinen zugeordnet. Hier wird vorteilhafterweise im zweiten Schreib/Lese-Speicher sehr wenig Speicherplatz benötigt, da allen Programmroutinen dieselben Speicherstellen zugeordnet sind. Es kann jedoch hierbei nur für alle  
5 Programmroutinen gleichermaßen durch entsprechende Belegung der zugeordneten Speicherstellen festgelegt werden, ob das jeweils einer Programmroutine zugeordnete Unterprogramm aufgerufen werden soll.

10

In einer besonders bevorzugten Ausführungsform wird durch den Parameter dem Unterprogramm die aufrufende Programmroutine mitgeteilt und davon abhängig die Funktion des Unterprogramms beeinflusst.

15

Weitere Vorteile, Merkmale und Anwendungsmöglichkeiten der Erfindung ergeben sich aus der nachfolgenden Beschreibung eines Ausführungsbeispiels in Verbindung mit der Zeichnung. In der Zeichnung zeigt

20

Figur 1 ein Ausführungsbeispiel der Erfindung; und

Figur 2 ein Ausführungsbeispiel einer Belegung des ersten und zweiten Schreib/Lese-Speichers.

25

In Figur 1 ist ein Festwertspeicher 1 dargestellt, in dem ein Programm abgelegt ist. Das Programm weist eine Vielzahl von Programmroutinen (im folgenden Routinen genannt) auf, wobei in Figur 1 nur drei Routinen 4, 5 und 6 skizziert sind. Durch  
30 Punkte und die Bezeichnung der Routinen mit Routine #1 bis Routine #n ist angedeutet, daß mehr als drei Routinen im Festwertspeicher 1 abgelegt sein können. Diese Routinen sind in das Programm in bestimmten vorgegebenen Adreßabständen "eingebaut". Wichtig ist dabei, daß die Routinen über den ge-  
35 samten Adreßbereich des Programms "gleichmäßig verstreut"



liegen, um jeden Teil des Programms nachträglich änderbar zu machen. Sind beispielsweise nur Routinen im niedrigen Adreßbereich des Programms vorhanden, kann vom höheren Adreßbereich des Programms nicht mehr in Unterprogramme gesprungen werden und demnach in diesem Adreßbereich auch keine Korrekturen mehr durchführen. Da die Routinen 4, 5 und 6 im Festwertspeicher 1 abgelegt sind, ist eine nachträgliche Änderung der Routinen selbst nicht mehr möglich.

10 Jeder der Routinen 4 bis 6 ist ein Unterprogramm 7 bis 8 zugeordnet. Die Unterprogramme sind dabei in einem ersten Schreib/Lese-Speicher 2 abgelegt und können jederzeit nachträglich geändert werden. Der erste Schreib/Lese-Speicher 2 liegt in demselben Adreßraum wie der Festwertspeicher 1, in dem das Programm abgelegt ist. Das bedeutet, daß sowohl der  
15 erste Schreib/Lese-Speicher 2 als auch der Festwertspeicher 1 über denselben Adreßbus und Datenbus angesteuert werden und im Programmspeicherbereich liegen. Dabei kann beispielsweise der erste Schreib/Lese-Speicher 2 den Adreßraum von Hex 0000  
20 bis Hex 1FFF und der Festwertspeicher 1 den Adreßraum von Hex 2000 bis Hex FFFF belegen. Der Schreib/Lese-Speicher 1 kann in diesem Fall durch ein RAM mit 8192 Byte realisiert werden.

In einem zweiten Schreib/Lese-Speicher 3 sind jeder Routine  
25 jeweils mehrere Speicherstellen zugeordnet. Der zweite Schreib/Lese-Speicher 3 liegt in einem anderen Adreßraum als der Festwertspeicher 1 und der erste Schreib/Lese-Speicher 2. Beispielsweise kann der zweite Schreib/Lese-Speicher 3 im Adreßraum eines Datenspeichers liegen. In Figur 1 ist dargestellt, daß der Routine 4 und 6 dieselben Speicherstellen 10  
30 zugeordnet sind, während der Routine 5 die Speicherstellen 9 zugeordnet sind. Die gestrichelte Linie zeigt, daß der Routine 5 auch die den Routinen 4 und 6 zugeordneten Speicherstellen 10 zugeordnet sein können. In diesem Fall sind allen Routinen  
35 4 bis 6 dieselben Speicherstellen zugeordnet, wodurch

15.12.98

10

der Speicherplatzbedarf im zweiten Schreib/Lese-Speicher 3 niedrig ist.

Bei der Zuordnung der Routinen zu den Unterprogrammen können  
5 drei verschiedene Fälle unterschieden werden, deren Vor- und Nachteile in der folgenden Tabelle verdeutlicht sind:

|   | Zuordnung   | Vorteile  | Nachteile  |
|---|---|---|--|
| 1 | ein Unterprogramm ist ausschließlich einer Routine zugeordnet: es gibt gleich viele Unterprogramme wie Routinen | Unterprogramm klein: es muß nur den Fall der aufrufenden Routine abhandeln        | zweiter Schreib/Lese-Speicher groß: Größe = "Anzahl der Unterprogramme" mal "pro Unterprogramm vorgesehene Speicherzellen" |
| 2 | ein Unterprogramm ist mehreren Routinen zugeordnet: es gibt weniger Unterprogramme als Routinen                 | Unterprogramm mittelgroß: es muß den Fall mehrerer aufrufender Routinen abhandeln | zweiter Schreib/Lese-Speicher mittelgroß   |
| 3 | ein Unterprogramm ist allen Routinen zugeordnet: es gibt eine Vielzahl von Routinen und ein Unterprogramm       | Unterprogramm groß: es muß den Fall jeder aufrufenden Routine abhandeln           | zweiter Schreib/Lese-Speicher klein  |

Im ersten Fall wird zwar die größte Vielseitigkeit erreicht,  
10 allerdings auf Kosten eines großen Speicherplatzbedarfs im zweiten Schreib/Lese-Speicher.

Im zweiten Fall erreicht man einen Ausgleich zwischen dem Speicherplatzbedarf im zweiten Schreib/Lese-Speicher und im  
15 ersten Schreib/Lese-Speicher und der Vielseitigkeit, da mehreren Routinen dieselben Speicherstellen im zweiten Schreib/Lese-Speicher und ein Unterprogramm zugeordnet ist. Das Unterprogramm bedient mehrere aufrufende Routinen und wird dementsprechend aufwendiger als ein Unterprogramm, das  
20 nur eine Routine bedient. Zur Unterscheidung der aufrufenden Routinen werden Parameter an das Unterprogramm übergeben.

Im dritten Fall gibt es nur ein Unterprogramm, das alle Routinen bedient und dementsprechend groß und aufwendig ist. Da-

15.12.98

für wird im zweiten Schreib/Lese-Speicher nur minimaler Speicherplatz benutzt.

Die Routinen 4 bis 6 weisen jeweils eine kurze Programmsequenz (Makro) auf, die aus dem zweiten Schreib/Lese-Speicher 3 Werte aus den der jeweiligen Routine zugeordneten Speicherzellen liest und mit einem vorgegebenen Wert vergleicht. Abhängig vom Ergebnis des Vergleichs ruft dann das Makro ein Unterprogramm auf, das in dem ersten Schreib/Lese-Speicher 2 abgelegt ist. Beim Aufruf des Unterprogramms wird ein Parameter an das Unterprogramm übergeben.

Ein Ausführungsbeispiel des Makros in einer Maschinensprache lautet beispielsweise folgendermaßen:

```
15 .MACRO      FW_HOOK_PAR HOOK_ID,HOOK_PAR
      mov     [##HK_XS.mem + HOOK_ID], a0
      brr     >%no_hook, eq
      mov     #HOOK_PAR, a1
20      call   a01

%no_hook:
.ENDM
```

(Aus dem zweiten Schreib/Lese-Speicher 3 wird der Inhalt der Adresse [##HK\_XS.mem + HOOK\_ID] in einen ersten Akkumulator a0 gelesen. Ist der gelesene Wert gleich Null, so wird an das Ende des Makros mit brr >%no\_hook, eq gesprungen. Ein Unterprogramm wird hierbei nicht aufgerufen. Andernfalls wird der Parameter HOOK\_PAR in einen zweiten Akkumulator a1 geschrieben und mit call a01 das dem Makro zugeordnete Unterprogramm aufgerufen.)

Die in dem zweiten Schreib/Lese-Speicher 3 abgelegten Werte können den Startadressen der Unterprogramme entsprechen.

15.12.98

112

In Figur 2 sind im ersten Schreib/Lese-Speicher 2 die unteren Adressen Hex 0000 bis Hex 001F nicht mit Unterprogrammen belegt. Ein erstes Unterprogramm #1 beginnt bei der Adresse Hex 0020, ein zweites Unterprogramm #2 bei der Adresse Hex 0040 und ein letztes Unterprogramm #n bei der Adresse Hex 00F0.

Im zweiten Schreib/Lese-Speicher 3 sind direkt die Startadressen Hex 0020, Hex 0040 und Hex 00F0 der Unterprogramme abgelegt.

Eine Routine liest aus den zugeordneten Speicherstellen des zweiten Schreib/Lese-Speichers 3 die dort gespeicherte Adresse, vergleicht die Adresse mit Null und verwendet die Adresse, falls diese ungleich Null ist, direkt als Einsprungsadresse in das im ersten Schreib/Lese-Speicher 2 abgelegte Unterprogramm.

Ist einer Routine kein Unterprogramm zugeordnet, wird in die zugeordneten Speicherzellen des zweiten Schreib/Lese-Speichers 3 einfach Hex 0000 abgelegt.

Beim Aufruf eines Unterprogramms können einer oder mehrere Parameter übergeben werden. Dies ist insbesondere vorteilhaft, wenn mehreren Programmroutinen dieselben Speicherstellen zugeordnet sind, in denen die Adresse eines Unterprogramms abgelegt ist, und das Unterprogramm eine der aufrufenden Programmroutine entsprechende Funktion ausführen soll. In diesem Fall muß das aufgerufene Unterprogramm die aufrufende Programmroutine erkennen. Das Unterprogramm wertet dazu den oder die übergebenen Parameter aus, wobei jeder Programmroutine, die das Unterprogramm aufrufen kann, ein bestimmter Parameterwert fest zugeordnet ist.

19.12.98

13

Vorteilhaft läßt sich das geschilderte Verfahren insbesondere bei Harvard-Architekturen, die vorwiegend bei digitalen Signalprozessoren (DSP = Digital Signal Processor) zum Einsatz kommen und getrennte Adreßräume für Programme und Daten aufweisen, anwenden.

19.12.98

14

## Patentansprüche

1. Verfahren zur Änderung des Ablaufs eines in einem Festwertspeicher abgelegten Programms, wobei das Programm eine  
5 Vielzahl von Programmroutinen (4 - 6) aufweist und wobei jeder Programmroutine (4 - 6) ein Unterprogramm (7, 8), das in einem ersten Schreib/Lese-Speicher (2) abgelegt ist, zuordenbar ist,

dadurch gekennzeichnet, daß

- 10 - jeder Programmroutine (4 - 6) Speicherstellen (9, 10) eines zweiten Schreib/Lese-Speichers (3) zugeordnet sind, und  
- eine Programmroutine (4 - 6), falls dieser Programmroutine (4 - 6) ein Unterprogramm (7, 8) zugeordnet ist, das Unterprogramm (7, 8) abhängig vom Inhalt der zugeordneten Speicherstellen (9, 10) aufruft.
- 15

2. Verfahren nach Anspruch 1,

dadurch gekennzeichnet, daß

- jede Programmroutine (4 - 6) beim Aufruf des ihr zugeordneten  
20 Unterprogramms (7, 8) mindestens einen Parameter an das Unterprogramm (7, 8) übergibt.

3. Verfahren nach Anspruch 1 oder 2,

dadurch gekennzeichnet, daß

- 25 die einer Programmroutine (5) zugeordneten Speicherstellen (9) ausschließlich dieser Programmroutine (5) zugeordnet sind.

4. Verfahren nach Anspruch 1 oder 2,

30 dadurch gekennzeichnet, daß

- die einer Programmroutine (4) zugeordneten Speicherstellen (10) auch allen anderen Programmroutinen (5, 6) zugeordnet sind.

- 35 5. Verfahren nach einem der Ansprüche 2 bis 4,

15.12.98

15

dadurch gekennzeichnet, daß  
durch den Parameter dem Unterprogramm (7, 8) die aufrufende  
Programmroutine (4 - 6) mitgeteilt wird und abhängig von dem  
Parameter die Funktion des Unterprogramms (7, 8) beeinflusst  
5 wird.

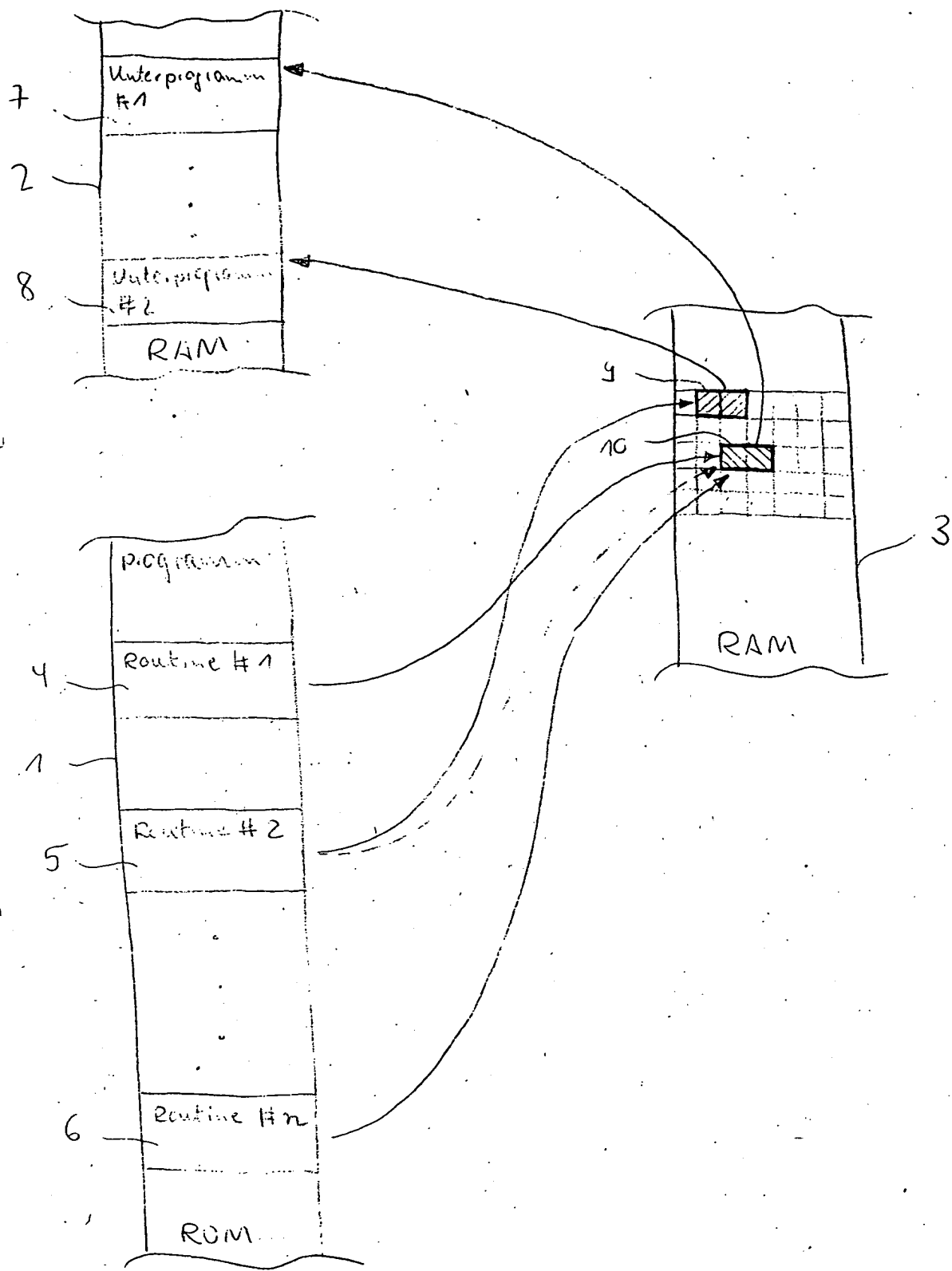


Fig. 1



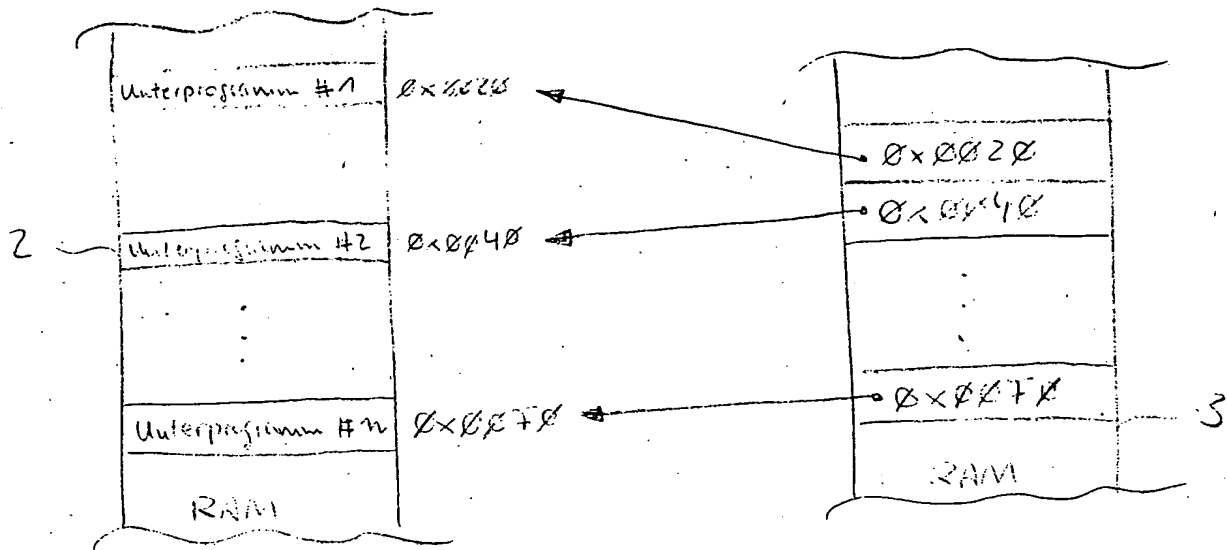


Fig. 2